Part 1

The "matplotlib" is a useful library for plotting curves and images in Python coding. This comprehensive library is an inspiration from "MATLAB" syntax for plotting figures.

There are two images named "Lenna256" and "Synth" that are shown in a single figure using "matplotlib" libraries.

Figure 1. Main images are depicted in a single figure using matplotlib library

Part 2

In this part, the discrete Fourier transform of main images is computed using "NumPy" built-in FFT. As the FFT results are imaginary, they are separated into magnitude and phase. So, every picture has two images as output. The results are as follows:

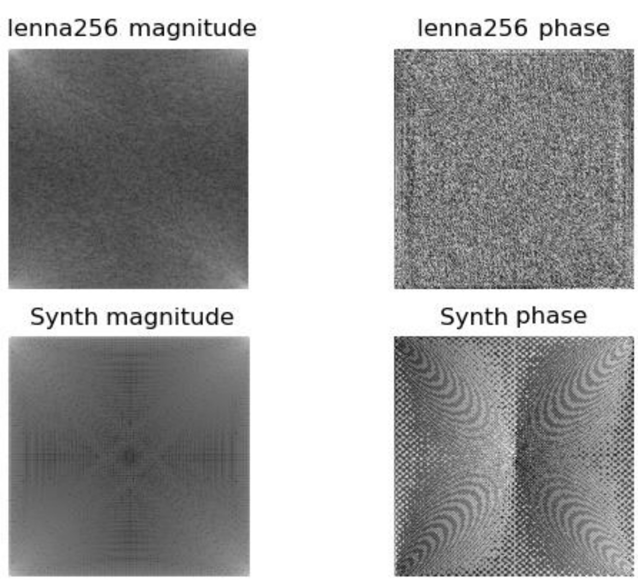

Figure 2. Discrete Fourier transform of main images using NumPy built-in FFT

Part 3

As Fourier transform result has a periodic form, the result of the transformation of images consists of two half parts. So, it is necessary to move the $F(0,0)$ (origin of Fourier transform function) to the center of the result. Therefore, there will be four quarter periods that meet in the center of the result. This translation can be done using the following formula:

$$f(x,y)(-1)^{x+y} \overset{DFT}{\Longleftrightarrow} F(u-\frac{M}{2}, v-N/2)$$

The output of the new shifted FFT is as follows:

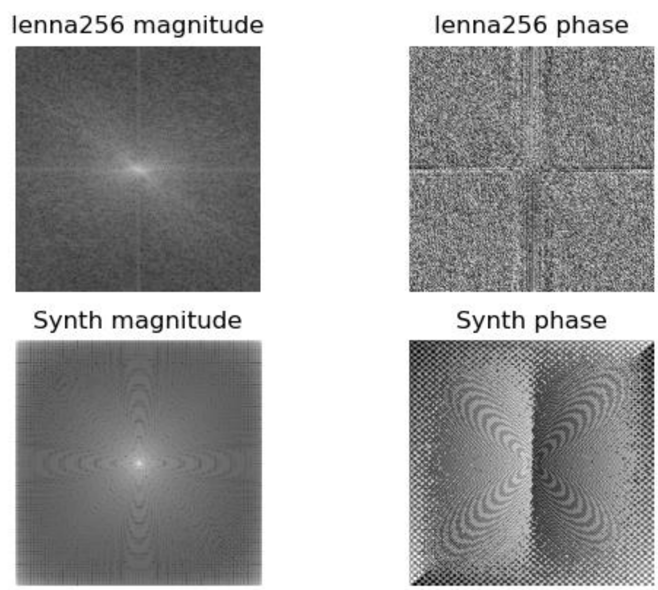

Figure 3. Shifted discrete Fourier transform of main images. The center of the images shows the origin of Fourier transform ($F(0,0)$).

The type of coordinate values in the spatial domains is integer. As images are 2-D discrete signals, they can be saved on storages with discrete numbers as coordinates. In contrast with this, the coordinate values in frequency domains are float, for signals can contain any float number of frequencies.

Part 4

Here, the magnitude and phase obtained in Part 2 are used to reconstruct new images by applying the inverse Fourier transform. Obviously, the result of applying inverse Fourier transform to the Fourier transform of the image is will result in the image itself. The result of reconstructing the image using magnitude and phase is as follows:

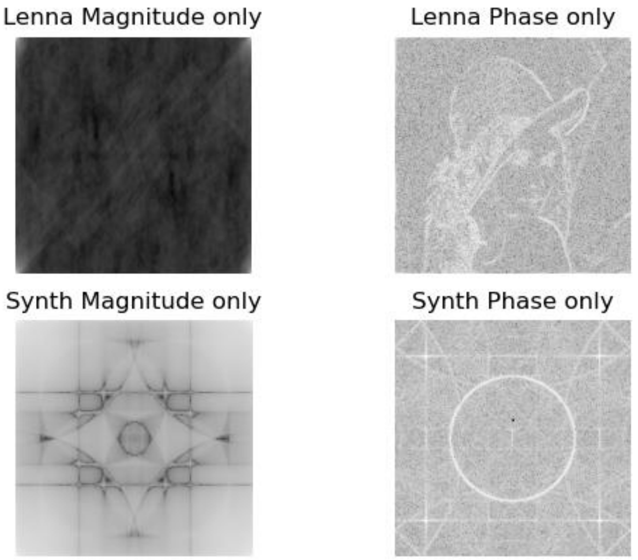

Figure 4. Inverse Fourier transform of the phase contains more information about the main image.

As it can be inferred from the output images, the inverse DFT of the phase carries more information than the magnitude. This image mostly depicts the edges and the shape of the objects in the image.

The inverse Fourier transform of the magnitude doesn't contain anything like the main image. Obviously, the combination of magnitude and phase results in the input image.

Part 5

Inverse Fourier transform of the combination of the phase of one image and the magnitude of another image gives the following outputs:

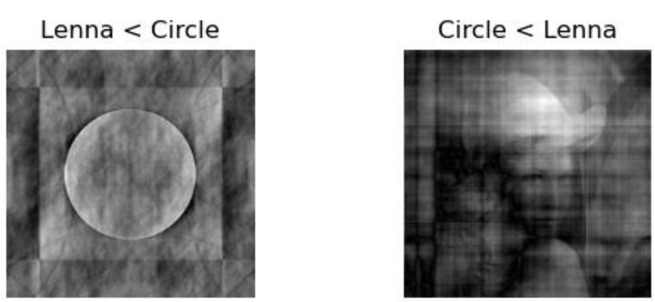

Figure 5. Reconstructing images using a combination of phase and magnitude of two images. In the title above the images, the name after sign '<' shows the name of the image which its phase used to reconstruct the new image.

The phase contains more information on the main image. Therefore, in the new image (that is reconstructed using the magnitude of one image and the phase spectra of another image), phase makes the shape of the new image. On the other side, the magnitude gives a new texture to the new image.

In figure 5, the new 'synth' image has a new texture that belongs to a scene, containing a face and walls and furniture. The new 'lenna256' image is simpler and smoother like the 'synth' image.

Part 6

In this part, an exponential term is multiplied by the Fourier transform of the images. This exponential term can simply shift the image to new coordinates.

$$F(u,v)e^{-j2\pi\left(\frac{ux_0}{M}+\frac{vy_0}{N}\right)} \Leftrightarrow f(x-x_0, y-y_0)$$

Using the above transformation, the images are shifted to M/2 along the x-axis and N/2 along the y-axis (by considering M and N as width and height of the image respectively).

The following images are the result:

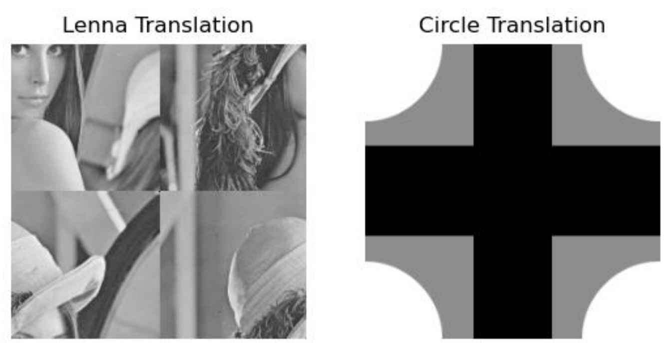

Figure 6. The images are divided to four parts and shifted as explained before.